JASPERREPORTS - REPORT VARIABLES

Report variables are special objects built on top of a report expression. Report variables simplify the following tasks:

- Report expressions which are heavily used throughout the report template. These expressions can be declared only once by using the report variables.
- Report variables can perform various calculations based on the corresponding expressions values like: count, sum, average, lowest, highest, variance, etc

If variables are defined in a report design, then these can be referenced by new variables in the expressions. Hence the order in which the variables are declared in a report design is important.

Variable Declaration

A variable declaration is as follows:

```
<variable name="CityNumber"
   incrementGroup="CityGroup" calculation="Count">
   <variableExpression>
       <![CDATA[Boolean.TRUE]]>
   </variableExpression>
</variable>
```

As seen above, <variable> element contains number of attributes. These attributes are summarized below:

The Name Attribute

Similar to *parameters* and *fields*, the *name* attribute of </variable> element is mandatory. It allows referencing the variable by its declared name in report expressions.

The Class Attribute

The *class* attribute is also mandatory and it specifies the class name for the variable values. Its default value is *java.lang.String*. This can be changed to any class available in the classpath, both at report-compilation time and report filling time. Irrespective of the type of a report value, the engine takes care of casting in the report expressions in which the $V{} token is used, hence making manual casts unnecessary.

Calculation

This attribute determines what calculation to perform on the variable when filling the report. The following subsections describe all the possible values for the calculation attribute of the <variable> element.

- *Average:* The variable value is the average of every non-null value of the variable expression. Valid for numeric variables only.
- *Count:* The variable value is the count of non-null instances of the variable expression.
- *First:* The variable value is the value of the first instance of the variable expression. Subsequent values are ignored.
- *Highest:* The variable value is the highest value for the variable expression.
- *Lowest:* The variable value is the lowest value for the variable expression in the report.

- *Nothing:* No calculations are performed on the variable.
- *StandardDeviation:* The variable value is the standard deviation of all non-null values matching the report expression. Valid for numeric variables only.
- *Sum:* The variable value is the sum of all non-null values returned by the report expression.
- *System:* The variable value is a custom calculation.(calculating the value for that variable yourself, using the scriptlets functionality of JasperReports)
- *Variance:* The variable value is the variance of all non-null values returned by evaluation of a report variable's expression.

Incrementer FactoryClass

This attribute determines the class used to calculate the value of the variable when filling the current record on the report. Default value would be any class implementing **net.sf.jasperreports.engine.fill.JRIncrementerFactory**. The factory class will be used by the engine to instantiate incrementer objects at runtime depending on the *calculation* attribute set for the variable.

IncrementType

This determines when to recalculate the value of the variable. This attribute uses values, as below:

- *Column:* The variable value is recalculated at the end of each column
- *Group:* The variable value is recalculated when the group specified by incrementGroup changes.
- *None:* The variable value is recalculated with every record.
- *Page:* The variable value is recalculated at the end of every page.
- *Report:* The variable value is recalculated once, at the end of the report.

IncrementGroup

This determines the name of the group at which the variable value is recalculated, when *incrementType* is *Group*. This takes name of any group declared in the JRXML report template.

ResetType

This determines when the value of a variable is reset. This attribute uses values, as below:

- *Column:* The variable value is reset at the beginning of each column.
- *Group:* The variable value is reset when the group specified by incrementGroup changes.
- *None:* The variable value is never reset.
- *Page:* The variable value is reset at the beginning of every page.
- *Report:* The variable value is reset only once, at the beginning of the report.

ResetGroup

This determines the name of the group at which the variable value is reset, when *resetType* is *Group*. The values for this attribute would be the name of any group declared in the JRXML report template.

Built-In Report Variables

There are some built-in system variables, ready to use in expressions, as follows:

Variable Name	Description
PAGE_NUMBER	This variable's value is its current page number. It can be used to display both the current page number and the total number of pages using a special feature of JasperReports text field elements, the *evaluationTime* attribute.
COLUMN_NUMBER	This variable contains the current column number
REPORT_COUNT	This report variable contains the total number of records processed.
PAGE_COUNT	This variable contains the number of records that were processed when generating the current page.
COLUMN_COUNT	This variable contains the number of records that were processed when generating the current column.
GroupName_COUNT	The name of this variable is derived from the name of the group it corresponds to, suffixed with the _COUNT sequence. This variable contains the number of records in the current group.

Example

Let's add a variable (**countNumber**) to our existing report template (Chapter Report designs). We will prefix the count to each record. The revised report template (jasper_report_template.jrxml) is as follows. Save it to C:\tools\jasperreports-5.0.1\test directory:

```
<?xml version="1.0"?>
<!DOCTYPE jasperReport PUBLIC
"//JasperReports//DTD Report Design//EN"
"http://jasperreports.sourceforge.net/dtds/jasperreport.dtd">

<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports
http://jasperreports.sourceforge.net/xsd/jasperreport.xsd"
name="jasper_report_template" pageWidth="595"
pageHeight="842" columnWidth="515"
leftMargin="40" rightMargin="40" topMargin="50" bottomMargin="50">
<parameter name="ReportTitle" />
<parameter name="Author" />

   <queryString>
    <![CDATA[]]>
   </queryString>

   <field name="country" >
      <fieldDescription>
         <![CDATA[country]]>
      </fieldDescription>
   </field>

   <field name="name" >
      <fieldDescription>
```

```
      <![CDATA[name]]>
   </fieldDescription>
</field>
<variable name="countNumber" >
   <variableExpression>
      <![CDATA[Boolean.TRUE]]>
   </variableExpression>
</variable>
<title>
   <band height="70">
      <line>
         <reportElement x="0" y="0" width="515"
         height="1"/>
      </line>
      <textField isBlankWhenNull="true" bookmarkLevel="1">
         <reportElement x="0" y="10" width="515"
         height="30"/>
         <textElement textAlignment="Center">
            <font size="22"/>
         </textElement>
         <textFieldExpression >
            <![CDATA[$P{ReportTitle}]]>
         </textFieldExpression>
         <anchorNameExpression>
            <![CDATA["Title"]]>
         </anchorNameExpression>
      </textField>
      <textField isBlankWhenNull="true">
         <reportElement  x="0" y="40" width="515" height="20"/>
         <textElement textAlignment="Center">
              <font size="10"/>
         </textElement>
         <textFieldExpression >
            <![CDATA[$P{Author}]]>
         </textFieldExpression>
      </textField>
   </band>
</title>

<columnHeader>
   <band height="23">
      <staticText>
         <reportElement mode="Opaque" x="0" y="3"
         width="535" height="15"
         backcolor="#70A9A9" />
         <box>
         <bottomPen lineWidth="1.0"
         lineColor="#CCCCCC" />
         </box>
         <textElement />
         <text><![CDATA[]]>
         </text>
      </staticText>
      <staticText>
         <reportElement x="414" y="3" width="121"
         height="15" />
         <textElement textAlignment="Center"
         verticalAlignment="Middle">
             <font isBold="true" />
         </textElement>
         <text><![CDATA[Country]]></text>
      </staticText>
      <staticText>
         <reportElement x="0" y="3" width="136"
         height="15" />
         <textElement textAlignment="Center"
         verticalAlignment="Middle">
            <font isBold="true" />
         </textElement>
         <text><![CDATA[Name]]></text>
      </staticText>
   </band>
</columnHeader>
```

```
   <detail>
      <band height="16">
         <staticText>
            <reportElement mode="Opaque" x="0" y="0"
            width="535" height="14"
            backcolor="#E5ECF9" />
            <box>
               <bottomPen lineWidth="0.25"
               lineColor="#CCCCCC" />
            </box>
            <textElement />
            <text><![CDATA[]]>
            </text>
         </staticText>
         <textField>
            <reportElement x="414" y="0" width="121"
            height="15" />
            <textElement textAlignment="Center"
            verticalAlignment="Middle">
               <font size="9" />
            </textElement>
            <textFieldExpression >
            <![CDATA[$F{country}]]>
            </textFieldExpression>
         </textField>
         <textField>
            <reportElement x="0" y="0" width="136"
            height="15" />
            <textElement textAlignment="Center"
            verticalAlignment="Middle" />
            <textFieldExpression >
            <![CDATA["  " + String.valueOf($V{countNumber}) +"."+$F{name}]]>
            </textFieldExpression>
         </textField>
      </band>
   </detail>
</jasperReport>
```

The java codes for report filling remains unchanged. The contents of the file **C:\tools\jasperreports-5.0.1\test\src\com\tutorialspoint\JasperReportFill.java** are as below.

```
package com.tutorialspoint;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.data.JRBeanCollectionDataSource;

public class JasperReportFill {
   @SuppressWarnings("unchecked")
   public static void main(String[] args) {
      String sourceFileName =
      "C://tools/jasperreports-5.0.1/test/jasper_report_template.jasper";

      DataBeanList DataBeanList = new DataBeanList();
      ArrayList<DataBean> dataList = DataBeanList.getDataBeanList();

      JRBeanCollectionDataSource beanColDataSource =
      new JRBeanCollectionDataSource(dataList);

      Map parameters = new HashMap();
      /**
       * Passing ReportTitle and Author as parameters
       */
      parameters.put("ReportTitle", "List of Contacts");
      parameters.put("Author", "Prepared By Manisha");
```

```
      try {
         JasperFillManager.fillReportToFile(
         sourceFileName, parameters, beanColDataSource);
      } catch (JRException e) {
         e.printStackTrace();
      }
   }
}
```

The contents of the POJO file **C:\tools\jasperreports-5.0.1\test\src\com\tutorialspoint\DataBean.java** are as below:

```
package com.tutorialspoint;

public class DataBean {
   private String name;
   private String country;

   public String getName() {
      return name;
   }

   public void setName(String name) {
      this.name = name;
   }

   public String getCountry() {
      return country;
   }

   public void setCountry(String country) {
      this.country = country;
   }
}
```

The contents of the file **C:\tools\jasperreports-5.0.1\test\src\com\tutorialspoint\DataBeanList.java** are as below:

```
package com.tutorialspoint;

import java.util.ArrayList;

public class DataBeanList {
   public ArrayList<DataBean> getDataBeanList() {
      ArrayList<DataBean> dataBeanList = new ArrayList<DataBean>();

      dataBeanList.add(produce("Manisha", "India"));
      dataBeanList.add(produce("Dennis Ritchie", "USA"));
      dataBeanList.add(produce("V.Anand", "India"));
      dataBeanList.add(produce("Shrinath", "California"));

      return dataBeanList;
   }

   /**
    * This method returns a DataBean object,
    * with name and country set in it.
    */
   private DataBean produce(String name, String country) {
      DataBean dataBean = new DataBean();
      dataBean.setName(name);
      dataBean.setCountry(country);
      return dataBean;
   }
}
```

Report Generation

We will compile and execute the above file using our regular ANT build process. The contents of the file build.xml (saved under directory C:\tools\jasperreports-5.0.1\test) are as below.

> *The import file - baseBuild.xml is picked from chapter Environment Setup and should be placed in the same directory as the build.xml.*

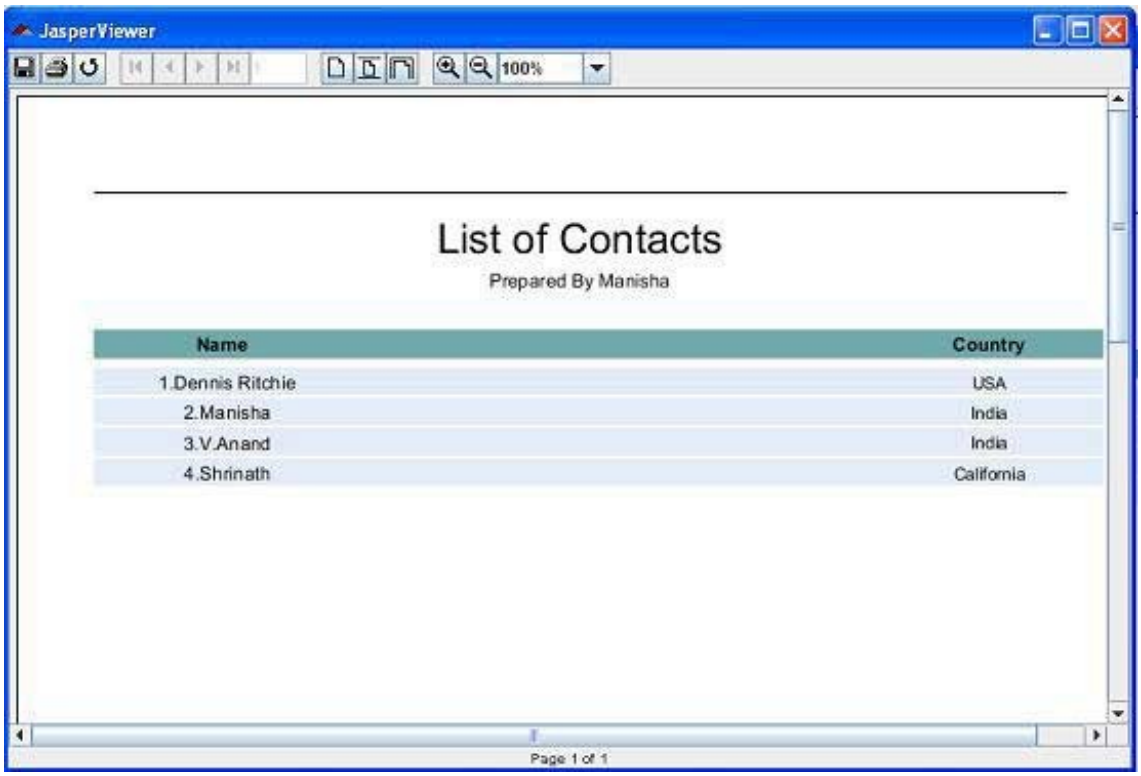
```
<?xml version="1.0" encoding="UTF-8"?>
<project name="JasperReportTest" default="viewFillReport" basedir=".">
   <import file="baseBuild.xml" />
   <target name="viewFillReport"
      depends="compile,compilereportdesing,run"
      description="Launches the report viewer to preview
      the report stored in the .JRprint file.">
      <java classname="net.sf.jasperreports.view.JasperViewer"
      fork="true">
         <arg value="-F${file.name}.JRprint" />
         <classpath ref />
      </java>
   </target>
   <target name="compilereportdesing"
      description="Compiles the JXML file and
      produces the .jasper file.">
      <taskdef name="jrc"
      classname="net.sf.jasperreports.ant.JRAntCompileTask">
         <classpath ref />
      </taskdef>
      <jrc destdir=".">
         <src>
         <fileset dir=".">
            <include name="*.jrxml" />
         </fileset>
         </src>
         <classpath ref />
      </jrc>
   </target>
</project>
```

Next, let's open command line window and go to the directory where build.xml is placed. Finally execute the command **ant -Dmain-class=com.tutorialspoint.JasperReportFill** (viewFullReport is the default target) as follows:

```
C:\tools\jasperreports-5.0.1\test>ant -Dmain-class=com.tutorialspoint.JasperReportFill
Buildfile: C:\tools\jasperreports-5.0.1\test\build.xml

clean-sample:
   [delete] Deleting directory C:\tools\jasperreports-5.0.1\test\classes
   [delete] Deleting: C:\tools\jasperreports-5.0.1\test\jasper_report_template.jasper
   [delete] Deleting: C:\tools\jasperreports-5.0.1\test\jasper_report_template.jrprint

compile:
    [mkdir] Created dir: C:\tools\jasperreports-5.0.1\test\classes
    [javac] C:\tools\jasperreports-5.0.1\test\baseBuild.xml:28: warning:
    'includeantruntime' was not set, defaulting to build.sysclasspath=last;
    set to false for repeatable builds
    [javac] Compiling 7 source files to C:\tools\jasperreports-5.0.1\test\classes

compilereportdesing:
      [jrc] Compiling 1 report design files.
      [jrc] log4j:WARN No appenders could be found for logger
      (net.sf.jasperreports.engine.xml.JRXmlDigesterFactory).
      [jrc] log4j:WARN Please initialize the log4j system properly.
      [jrc] log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig
      for more info.
      [jrc] File : C:\tools\jasperreports-5.0.1\test\jasper_report_template.jrxml ... OK.

run:
     [echo] Runnin class : com.tutorialspoint.JasperReportFill
     [java] log4j:WARN No appenders could be found for logger
     (net.sf.jasperreports.extensions.ExtensionsEnvironment).
     [java] log4j:WARN Please initialize the log4j system properly.
```

```
viewFillReport:
     [java] log4j:WARN No appenders could be found for logger
     (net.sf.jasperreports.extensions.ExtensionsEnvironment).
     [java] log4j:WARN Please initialize the log4j system properly.

BUILD SUCCESSFUL
Total time: 18 seconds
```

As a result of above compilation, a JasperViewer window opens up as in the screen below:



Here we see that count is prefixed for each record.